

# Comparison of iterative solvers for electromagnetic analysis of plasmonic nanostructures using multiple surface integral equation formulations

Hipólito Gómez-Sousa<sup>1</sup>, Óscar Rubiños-López<sup>1</sup> and José Ángel Martínez-Lorenzo<sup>2</sup>

<sup>1</sup>Department of Signal Theory and Communications, University of Vigo,  
EI de Telecomunicación, ES 36310 Vigo, Spain.

[{hgomez, oscar}@com.uvigo.es](mailto:{hgomez, oscar}@com.uvigo.es)

<sup>2</sup>Northeastern University, 360 Huntington Ave.,  
Suite 302 Stearns Center, Boston, MA 02115, USA.

[jmartinez@coe.neu.edu](mailto:jmartinez@coe.neu.edu)

## Abstract

The electromagnetic behavior of plasmonic structures can be predicted after discretizing and solving a linear system of equations, derived from a continuous surface integral equation (SIE) and the appropriate boundary conditions, using a method of moments (MoM) methodology. In realistic large-scale optical problems, a direct inversion of the SIE–MoM matrix cannot be performed due to its large size, and an iterative solver must be used instead. This paper investigates the performance of four iterative solvers (GMRES, TFQMR, CGS, and BICGSTAB) for five different SIE–MoM formulations (PMCHWT, JMCIE, CTF, CNF, and MNMF). Moreover, under this plasmonic context, a set of suggested guidelines are provided to choose a suitable SIE formulation and iterative solver depending on the desired simulation error and available runtime resources.

## 1 Introduction

During the last years, we have witnessed an increasing interest in the optical properties of metallic nanoparticles. Research in the area of plasmonic nanoelectronics explores the behavior of electromagnetic fields which are confined over dimensions smaller than the wavelength. This field confinement is induced by interactions between electromagnetic waves and conduction electrons at the interfaces of metallic nanostructures. These plasmonic interactions can be used to manipulate light beyond the classical diffraction limit, giving rise to an emerging wide range of interesting novel applications in sensing and waveguiding, amongst other applications [1, 2]. Additionally, at optical frequencies, metallic nanostructures can be characterized by localized surface-plasmon resonances (LSPRs) that support a strong enhancement in the directivity of spontaneous emission of light by single fluorescent molecules or other point-like emitters of light. This enhancement in the emission of light is a fundamental aspect that has recently originated an intensive promising research work on the design and the experimental construction and testing of nanoantennas [3–7]. As a consequence of this novel research area pioneered in recent years, numerical techniques for providing accurate simulation and analysis of problems involving plasmonic nanostructures are indispensably required to exploit the growing range of applications relying on optical plasmonic properties.

The behavior of nanoparticles at optical frequencies can be well modeled by classical electrodynamics [1]. Nevertheless, in electromagnetic optics, the penetration of fields into metals must be considered and therefore SIE (surface integral equation) formulations for penetrable scatterers are a suitable choice for simulating the optical responses of isolated plasmonic nanoparticles or composite structures consisting of multiple nanoparticles. Surface integral equation techniques based on the method of moments (MoM) have demonstrated to provide very accurate simulation results in many different problems involving real plasmonic objects [8–11]. In spite of being a classical approach, the SIE–MoM method delivers accurate predictive results for particle and surface feature sizes down to  $\sim 1$  nm, a distance below which quantum non-local phenomena become non-negligible [11]. Although not yet widely employed in optics, the SIE–MoM approach brings important advantages over volumetric approaches such as the discrete-dipole approximation (DDA) [12], the finite difference in time domain (FDTD) method [13], and the frequency-domain finite-element (FEM) methods [14]. Such advantages include the fact that the SIE–MoM approach requires discretizations of the material boundary surfaces only, thus generally reducing the number of unknowns with respect to integral formulations based on volumetric mesh modeling. Additionally, absorbing boundary conditions and surrounding empty space need not to be parametrized, resulting in a significantly easier mesh modeling. Finally, the SIE–MoM methods are less prone to be affected by instabilities due to abrupt and spatially rapid variations of the permittivity [11], as commonly occurs in the case of plasmonic problems.

There are many known factors that determine the final accuracy and the total runtime when simulating with the SIE–MoM approach. However, two of them are of special interest when solving this type of problems: the SIE–MoM formulation, and the iterative numerical solver. Previous studies have concentrated on assessing which SIE formulations are more suitable for different types of electromagnetic scattering and radiation problems in terms of both runtime and accuracy [15–20], including problems involving plasmonic structures [20]. Nevertheless, there is a general lack of publications that focus explicitly on the role of the iterative solver. Addressing this lack of results would certainly be relevant as the selected iterative solver has a strong impact on the total simulation runtime and memory consumption. This impact is particularly important when dealing with real-world plasmonic problems, because they usually require performing massive batches of large-scale simulations for different varying parameters such as optical wavelength, illumination beam, and relative permittivity.

A comparative study of the performance of some well-known SIE formulations was first released in [15] for non-plasmonic conducting and dielectric objects. Later on [16], this was also done for perfect electric conductor (PEC) bodies. A similar study for dielectric objects, but using acceleration algorithms for MoM, was carried out in [17]. A comparative study applying the SIE–MoM approach to left-handed metamaterials (LHM’s) was reported in [18]; and, later on, this was done using acceleration algorithms [19]. The foremost published comparison of five widespread SIE formulations in the interesting context of plasmonic media was accomplished in [20]. All the above-mentioned publications also include iterative-performance studies. However, these studies are only focused on a single iterative solver. The iterative-performance study carried out in [15] was realized using only as iterative solver the ordinary (without restarts) version of the generalized minimal residual method (GMRES) [21]. A restarted GMRES with fixed restart parameter was chosen in [16, 18, 20], whereas another iterative solver called BICGSTAB [22] was used for all the analyses published in [17, 19].

As detailed in 1.2.5 of [23], the algorithmic performance and the memory complexities of each iterative solver may vary significantly depending on the considered type of engineering problem. As a consequence, in order to optimize the performance and memory usage for a given class of problems, there are no general selection rules for the solver and an investigation should be required to compare iterative solutions provided by various algorithms. This kind of important study was

left out in all the iterative-performance analyses reported in the aforesaid publications [15–20], as these papers perform comparative analyses by only varying the SIE formulation for a fixed iterative solver. In the present work, we extend to multiple iterative solvers some comparative multiple-SIE single-solver results published in [20] for plasmonic scatterers. Furthermore, a memory complexity analysis not included in the previously mentioned reference is presented in this paper. Moreover, the nanoscatterers used for obtaining the results in [20] consist only of spheres simulated at a single operating frequency. The selection of spheres as targets is well justified by the availability of the Mie’s series analytical reference results [24]; however, in the present paper we also analyze the iterative performance results when dealing with more elaborate geometries, such as a real plasmonic nanoantenna. In addition to the study of the iterative performance, we also check, following a procedure similar to that in [20], the accuracy of each SIE formulation when the MoM linear system is solved by different iterative techniques.

The SIE formulations considered for the iterative performance comparison in this paper include tangential equations only (combined tangential formulation, known as CTF, and Poggio-Miller-Chang-Harrington-Wu-Tsai formulation, named by the acronym PMCHWT), normal equations only (combined normal formulation, CNF, and modified normal Müller formulation, MNMF) and both normal and tangential equations (electric and magnetic current combined-field integral equation, JMCFIE). A detailed description of these five formulations can be found in [20, 23]. The four studied iterative methods, based on Krylov subspaces, used for solving linear systems of equations are the following: GMRES (generalized minimum residual method), TFQMR (transpose free quasi-minimal residual method), CGS (conjugate gradients squared method) and BICGSTAB (bi-conjugate gradients stabilized method). Thorough explanations for all these solvers can be found in [22]. In the particular case of GMRES, in addition to the iterative performance, we analyze the influence of the restart parameter on the memory consumption as well.

The rest of this paper is organized as follows. The SIE formulations considered in the study are described in Section 2. Next, in Section 3, a concise overview on Krylov iterative methods applied to SIE formulations is given. We present in Section 4 a comparative study of numerical results for the considered SIE formulations. First, in Subsection 4.1, representative sets of near and far field outcomes obtained employing each SIE formulation are compared with the analytical results provided by the Mie’s series, in order to determine the level of accuracy for each formulation. Then, in Subsection 4.2, the iterative performance of each linear solver is assessed by measuring the runtime to solve the MoM system and the total number of required matrix-vector multiplications for different plasmonic geometries. Finally, Section 5 concludes the paper with a summary.

## 2 Surface integral equation formulations

In the analysis of the scattered field produced by an impinging electromagnetic wave on a penetrable object, both tangential and normal boundary conditions are typically imposed for the electric and magnetic fields at the interface of the object. These boundary conditions establish the normal electric field integral equation (N–EFIE), the normal magnetic field integral equation (N–MFIE), the tangential electric field integral equation (T–EFIE) and the tangential magnetic field integral equation (T–MFIE). The following linear combinations of the aforementioned formulations are known to provide stable sets of SIE formulations [20, 23]:

$$\begin{aligned} \frac{a_1}{\eta_1} (\text{T} - \text{EFIE}_1) + \frac{a_2}{\eta_2} (\text{T} - \text{EFIE}_2) + b_1 (\text{N} - \text{MFIE}_1) - b_2 (\text{N} - \text{MFIE}_2) &= \vec{0}, \\ -c_1 (\text{N} - \text{EFIE}_1) + c_2 (\text{N} - \text{EFIE}_2) + d_1 \eta_1 (\text{T} - \text{MFIE}_1) + d_2 \eta_2 (\text{T} - \text{MFIE}_2) &= \vec{0}. \end{aligned} \quad (1)$$

In the preceding equations, we employ the same sign conventions as in [23].  $\eta_i = \sqrt{\mu_i/\varepsilon_i}$  is the intrinsic impedance in region  $R_i$  for  $i = 1, 2$ .  $R_1$  and  $R_2$  are the outer and inner regions of the

scatterer, respectively. Different values can be assigned to the complex scalar parameters  $a_i, b_i, c_i, d_i$  for  $i = 1, 2$  in order to obtain valid stable formulations. The expressions for all the identities involved in Eq. (1) are the following:

$$\begin{aligned} \text{T - EFIE}_i: \quad \vec{0} &= \mathbf{E}_i^{\text{inc}}(\mathbf{r}) \Big|_{\text{tan}} - L_i \mathbf{J}(\mathbf{r}) \Big|_{\text{tan}} + K_i \mathbf{M}(\mathbf{r}) \Big|_{\text{tan}} - (-1)^i \frac{1}{2} \mathbf{M}(\mathbf{r}) \times \hat{\mathbf{n}}(\mathbf{r}), \\ \text{T - MFIE}_i: \quad \vec{0} &= \mathbf{H}_i^{\text{inc}}(\mathbf{r}) \Big|_{\text{tan}} - K_i \mathbf{J}(\mathbf{r}) \Big|_{\text{tan}} - \frac{1}{\eta_i} L_i \mathbf{M}(\mathbf{r}) \Big|_{\text{tan}} + (-1)^i \frac{1}{2} \mathbf{J}(\mathbf{r}) \times \hat{\mathbf{n}}(\mathbf{r}), \\ \text{N - EFIE}_i &= \hat{\mathbf{n}}(\mathbf{r}) \times (\text{T - EFIE}_i), \quad \text{N - MFIE}_i = \hat{\mathbf{n}}(\mathbf{r}) \times (\text{T - MFIE}_i). \end{aligned} \quad (2)$$

In Eqs. (2),  $\mathbf{J}(\mathbf{r})$  and  $\mathbf{M}(\mathbf{r})$  denote the, a-priori unknown, induced equivalent surface currents (electric and magnetic currents respectively) on the interface between  $R_1$  and  $R_2$ .  $\mathbf{J}(\mathbf{r})$  and  $\mathbf{M}(\mathbf{r})$  are vector functions of an arbitrary surface point  $\mathbf{r}$ , which is defined approaching the surface from  $R_1$ . Vector  $\hat{\mathbf{n}}(\mathbf{r})$  is the unit normal to the surface, pointing towards exterior region  $R_1$ . Vectors  $\mathbf{E}_i^{\text{inc}}(\mathbf{r})$  and  $\mathbf{H}_i^{\text{inc}}(\mathbf{r})$  respectively represent the incident electric and magnetic fields at surface point  $\mathbf{r}$ .  $L_i$  and  $K_i$  are used to denote integro-differential operators defined as

$$\begin{aligned} L_i \mathbf{X}(\mathbf{r}) &= \int_S \left[ j\omega\mu_i \mathbf{X}(\mathbf{r}') + \frac{j}{\omega\varepsilon_i} \nabla (\nabla' \cdot \mathbf{X}(\mathbf{r}')) \right] G_i(\mathbf{r}, \mathbf{r}') ds', \\ K_i \mathbf{X}(\mathbf{r}) &= \oint_S \mathbf{X}(\mathbf{r}') \times \nabla G_i(\mathbf{r}, \mathbf{r}') ds'. \end{aligned} \quad (3)$$

The symbol  $\oint$  is used in the definition of  $K_i$  for indicating that the integration is taken as a Cauchy principal value integral. The integration surface  $S$  refers to the separation interface between  $R_1$  and  $R_2$ . The term  $G_i(\mathbf{r}, \mathbf{r}')$  in (3) refers to the scalar Green's function.

A generalization of the SIE–MoM formulation for the analysis of *multiple* plasmonic media can be looked up in [9], or a different alternative approach can be found in [25]. The discretization of the unknown currents into basis functions and the generation of the MoM linear system can be consulted, for example, in [9, 23]. The comparative study included in the following sections considers the five widespread formulations defined for the parameters  $a_i, b_i, c_i, d_i$  in Table 1.

Table 1: Parameters for obtaining five well-documented surface integral equation formulations.

<i>formulation</i>	$a_i$ for $i = 1, 2$	$b_i$ for $i = 1, 2$	$c_i$ for $i = 1, 2$	$d_i$ for $i = 1, 2$
<b>PMCHWT</b>	$\eta_i$	0	0	$1/\eta_i$
<b>JMCFIE</b>	1	1	1	1
<b>CTF</b>	1	0	0	1
<b>CNF</b>	0	1	1	0
<b>MNMF</b>	0	$\mu_i/(\mu_1 + \mu_2)$	$\varepsilon_i/(\varepsilon_1 + \varepsilon_2)$	0

### 3 Iterative method overview

The adequate choice of the iterative method to solve plasmonic problems involving a high number of unknowns is fully supported by the fact that MoM-based accelerating techniques –such as the fast multipole method (FMM) and the multilevel fast multipole algorithm (MLFMA)– are usually employed for problems involving hundreds of thousands of unknowns. Unlike pure MoM, these MoM-based accelerating techniques accept a controllable error in the solution [23]. In the FMM and MLFMA algorithms, the direct inversion of the MoM matrix is not feasible, even on modern parallel computers; and, for this reason, iterative solvers are indispensable to find the initially unknown

vector. Even when using a pure MoM code, without FMM and MLFMA accelerations, an iterative solver usually provides faster convergence than a direct solver such as LU matrix factorization. In fact, the computational complexity of pure MoM with a direct solver is  $O(N^3)$ , where  $N$  is the number of unknowns. This complexity can be easily lowered to  $O(N^2)$  by simply switching from a direct to an iterative solver [26].

Krylov iterative solvers are among the most popular in computational electromagnetics, because of their ability to deliver good rates of convergence and to efficiently handle very large problems [16]. This kind of methods look for the vector solution  $\mathbf{I}$  of the system  $\bar{\mathbf{Z}}\mathbf{I} = \mathbf{V}$  in the Krylov space  $K_n(\bar{\mathbf{Z}}, \mathbf{V}) = \text{span}\{\mathbf{V}, \bar{\mathbf{Z}}\mathbf{V}, \bar{\mathbf{Z}}^2\mathbf{V}, \dots, \bar{\mathbf{Z}}^{n-1}\mathbf{V}\}$ , where  $n = 1, 2, \dots$  represents a number of iteration in the solver.  $K_n(\bar{\mathbf{Z}}, \mathbf{V})$  is a suitable space from which one can construct approximate solutions of the linear system of equations, since it is closely related to  $\bar{\mathbf{Z}}^{-1}$  [16, 22].

When any programmed Krylov method is called, a tolerance value  $\eta$  is inputted to the code. In practice, the solver does not run until an exact solution is found, but rather terminate at iteration  $n$  if a certain criterion has been satisfied for the estimated solution  $\mathbf{I}_n$ . One typical criterion is to terminate the algorithm after the following inequality is met:

$$\|\bar{\mathbf{Z}}\mathbf{I}_n - \mathbf{V}\|_2 / \|\mathbf{V}\|_2 \leq \eta. \quad (4)$$

The term  $\|\bar{\mathbf{Z}}\mathbf{I}_n - \mathbf{V}\|_2 / \|\mathbf{V}\|_2$  is called relative residual. Typical values for the tolerance are  $\eta = 10^{-6}$  for double precision (64 bits) entries in  $\bar{\mathbf{Z}}$  and  $\mathbf{V}$ , and  $\eta = 10^{-3}$  for single precision (32 bits) entries. However, for MoM-based simulations involving tens of millions of unknowns, the tolerance must be raised to values as high as about  $\eta = 10^{-2}$  [27]. This increase in  $\eta$  is needed because the condition number of the MoM system typically worsens as the number of unknowns lifts and the number of iterations to achieve  $\eta$  would be impracticable to reach.

The Krylov method GMRES (generalized minimum residual method), developed by Y. Saad and H. Schultz in 1986 [21], is known to be the optimal iterative solver in the sense that it minimizes the number of iterations required to converge satisfying (4) [16]. Nevertheless, the optimality of GMRES comes at a price. The memory cost of applying the method increases with the iterations, and it may sometimes become prohibitive for solving certain problems. As an attempt to limit this cost, there exist restarted versions of GMRES in which, after a given number  $R$  of iterations ( $R$  is known as the restart parameter), the approximate solution for the next steps is computed from the previously generated Krylov subspace. Then this existent Krylov subspace is completely erased from the memory, and a new space is constructed from the latest residual. In a restarted GMRES, each ordinary iteration is called internal iteration, whereas a set of  $R$  ( $R$  denotes again the restart value) internal iterations is called external iteration. Summarizing the storage costs, the number of additional floats or doubles –over the baseline memory requirements in MoM/FMM/MLFMA– that GMRES requires in memory can be easily estimated from the following analytical results [28]:

$$GMES\_RAM\_overuse(M, N) = 2MN + M^2 - 7M \quad \text{floats or doubles}, \quad (5)$$

where  $N$  is the number of complex unknowns and  $M$  is the number of iterations required to achieve (4) in the unrestarted ordinary GMRES. For a restarted GMRES,  $M$  is computed from the restart parameter  $R$  as  $M = \min(P, R)$ , with  $P$  the total number of internal iterations.

In addition to the restarted versions of GMRES, other non-optimal iterative solvers attempt to preserve the favorable convergence properties of GMRES while introducing a negligible overuse in RAM usage. For this paper, we consider complex versions of iterative Krylov solvers based on different versions of the so-called Lanczos biorthogonalization algorithm [22]: TFQMR (transpose free quasi-minimal residual method), CGS (conjugate gradients squared method) and BICGSTAB (bi-conjugate gradients stabilized method).

## 4 Comparative study of numerical results

The analysis of the accuracy and the iterative performance of the five SIE formulations described in Section 2 is carried out in this section for representative problems involving plasmonic materials. Error and iterative performance results were obtained for the Krylov solvers described in Section 3. The error analyses are summarized in Subsection 4.1. Then, in Subsection 4.2, we present iterative performance analyses that not only include spheres as scatterers, but also real models of plasmonic nanoantennas. The GMRES solver with a fixed restart was the only solver considered in [20], whereas our results for plasmonic problems cover the following important cases: three additional solvers (TFQMR, CGS and BICGSTAB), the ordinary GMRES without restarts, and the restarted GMRES including a set of different restart parameters not considered in the aforesaid reference. Furthermore, the results in [20] cover up to 36,000 unknowns. In contrast, we have extended the comparative results up to around 100,000 unknowns.

Some quantitative results presented in this paper regarding the iterative performance and the accuracy differ from their counterparts in [20] –even though there is a general agreement among those main qualitative results that are common to both papers– due to two major reasons: i) we employed a Gaussian quadrature rule, described in [29], consisting of 7 points per triangle for numerical integration, whereas in [20] a 3-point rule was used; ii) unlike the simulations in [20], we employed a diagonal preconditioner described in [29] together with the preconditioning technique in [30]. A diagonal preconditioner and the technique in [30] are straightforward to implement in any existing MoM code without adding significant computational complexity. Moreover, in 10.2.4 of [29] is stated that experimental results show that the 7-point quadrature rule is preferable to the 3-point rule, as it provides superior precision for MoM problems, which strengthens our choice for this integration technique.

This paper also describes the numerical precision and meshes employed for the discretization, not clearly stated in some of the aforementioned literature, in order to allow a full reproducibility of our results by the scientific community. Double-precision floating-point C calculations were used for all the results in this paper, and we employed the so-called “frontal” mesh type in the free software Gmsh [31]. The “frontal” type was selected because it provides triangles with good aspect ratios, namely, triangles which do not have small internal angles. This triangle feature is required in MoM to obtain accurate results, as explained in 8.7.4.1 of [29].

For all the studied iterative solvers, the maximum number of external iterations was unlimited –with the exception of the unrestarted GMRES, where the number of iterations is limited to the number of unknowns–, and the relative residue tolerance for stopping each method was set to  $10^{-6}$ , in accordance with the traditionally used tolerance value which can be found, for instance in [20, 30].

The main graphical representations of the results in this paper were obtained with a pure MoM implementation, as in [20]. However, some figures in this paper include MLFMA results whenever a “MLFMA vs. pure MoM” comparison is relevant. The MLFMA implementation used for the present work was configured with the same MLFMA parameters described in [32]. Our computational implementation of the MoM/MLFMA-SIE method consists of a regular C code involving double-precision floating-point calculations. For the accurate evaluation of the singular integrals, we have used all the proper analytical extraction techniques explained in [29]. In order to model the MoM currents, we have employed the well-known Rao-Wilton-Glisson (RWG) basis functions [29]. The Galerkin’s method was assumed for this work, meaning that the MoM testing functions are the same as the basis functions.

All our simulations were carried out on a computer with two Intel® Xeon® Processors E5-2690v2 running the 64-bit operating system Windows 8.1 Professional. The code was not parallelized by hand, but automatically parallelized using the source-to-source compiler Parallware [33] to be run with 20 threads executed on half of the cores. The aforementioned model of the processors

and the number of threads have no influence on any comparative result, but only determine the absolute quantitative runtime outcomes in Subsection 4.2. Regarding the automatic parallelization, Parallware is a parallelizing tool that automatically extracts the parallelism implicit in the source code of a sequential simulation program written in the regular C programming language. In addition, Parallware automatically generates an optimized parallel-equivalent program written in C and annotated with OpenMP [34] compiler pragmas.

#### 4.1 Assessment of numerical accuracy

A comparative analysis of the accuracy was carried out employing the five selected MoM-based SIE formulations and the four solvers. The same sphere with radius  $\lambda_0/2$  employed in 3.3 of [20] was used in this section as a representative example for evaluating the accuracy of each considered iterative solver. This sphere is illuminated by a plane wave, with incidence direction  $\hat{\mathbf{z}}$  and polarization  $\hat{\mathbf{x}}$ , at the operating optical wavelength  $\lambda_0 = 548.6$  nm. The following plasmonic materials were chosen for the sphere composition: gold ( $\varepsilon_r = -5.8 - j2.1$ , at the simulation frequency), silver ( $\varepsilon_r = -12.8 - j0.4$ ) and aluminum ( $\varepsilon_r = -35.2 - j9.82$ ). The values of the complex relative dielectric permittivity for each material have been extracted from [20].

The normalized root mean square (RMS) error with respect to the Mie's series results was calculated using the following expression:

$$e_{rms} = \left( \sqrt{\sum (E_{ref} - E_{simul})^2 / N} \right) / \max \{E_{ref}\}, \quad (6)$$

where  $E_{simul}$  is the magnitude of the scattered electric far field obtained in the simulation for  $N$  different observation values.  $E_{ref}$  is the reference field provided by the Mie's series. For far-field patterns, the error was calculated using  $N = 360$  equispaced angular values for the variable  $\phi$  on plane XZ. The near-field patterns comprise a mesh with resolution  $200 \times 200$  points. This mesh was created on plane XZ over a centered square of side length  $4r$ , with  $r$  the sphere radius.

We found no significant differences in the comparative error variation among the three considered plasmonic materials. We did not either encounter any effect of the chosen iterative solver on the normalized RMS error, as all the four solvers yielded identical error levels. These results in this section complete the important findings published in [20], extending the error analysis to multiple iterative solvers. Fig. 1 shows the error versus the number of unknowns for each SIE formulation when the sphere is made of gold. The variation on the number of unknowns was achieved by varying the maximum side length  $\ell_{discr}$  of the triangles in the geometry mesh, according to the rule  $\ell_{discr} = \frac{\lambda_0}{10n}$ ,  $n = 1, 2, \dots, 7$ .

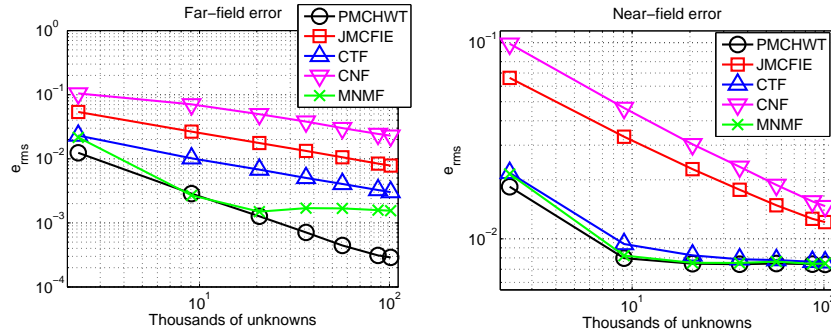


Figure 1: Comparative study of the RMS error for a golden sphere with radius  $r = \lambda_0/2$ : (left) far field; (right) near field. Error values were found to be identical among the four considered iterative solvers.

In terms of accuracy, the PMCHWT has clearly shown to be the most reliable formulation in plasmonics. This statement is in full agreement with the results in [20]. The MNMF formulation has an error level comparable to the PMCHWT in most cases; however its far-field error level becomes unstable and starts to grow for high numbers of unknowns. This anomalous behavior in MNMF might be ascribed to the singularity extraction techniques involved in the MNMF implementation [29]. The CTF formulation exhibits a near-field error similar to the PMCHWT, but in the far field its error level is notably worse. Given that both PMCHWT and CTF combine tangential equations only, the accuracy provided by the PMCHWT highlights the actual importance of the combination scalar parameters on the obtained level of accuracy.

As a graphical reference, Fig. 2 shows far and near-field reference values obtained from the Mie's series for the sphere with radius  $\lambda_0/2$ . PMCHWT far-field values simulated with two very different numbers of unknowns (mesh sizes) are represented in Fig. 2 in order to allow for visual assessment of the real importance of taking into account the error variations shown in Fig. 1.

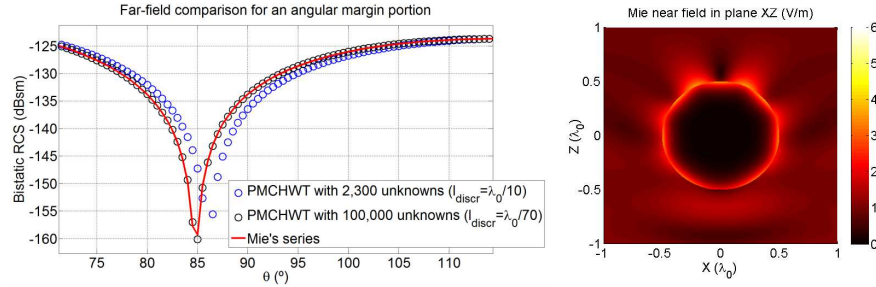


Figure 2: Reference results from Mie's series: (left) far field; (right) near field. The far-field results include a comparison with PMCHWT predictions for two different representative mesh sizes.

## 4.2 Assessment of iterative performance

With the purpose of assessing the iterative performance of the five studied SIE formulations, we initially simulated the  $\lambda_0/2$  radius gold sphere, using MoM solved iteratively. We employed four well-known iterative solvers for linear systems of equations [22]: TFQMR, CGS, BICGSTAB and GMRES –an unrestarted GMRES version, plus three more GMRES versions with restart parameters 30, 60 and 90–. In Fig. 3, we analyze the influence of the GMRES restart parameter on the memory cost. This analysis is used to justify our elections for the mentioned restart values. Fig. 3 shows on the left the memory usage required by pure MoM and also by MLFMA, as a function of the number of unknowns. These curves on the left are independent of the particular chosen solver. Then, on the right of Fig. 3, the RAM overuse introduced by GMRES is represented for the restart values 30, 60 and 90, and for the unrestarted version.

As inferred from Fig. 3, the percentage RAM overuse may be negligible when using pure MoM together with unrestarted GMRES, but is noticeably worse in the MLFMA case. In other words, even if the available RAM allows a MLFMA simulation, the election of the unrestarted GMRES for the solver may make this simulation unfeasible. The maximum GMRES restart considered in the present work,  $R=90$ , was selected to bound the total RAM consumption increase in MLFMA to around 1% of the total RAM usage in every non-GMRES Krylov iterative solver.

Measurements of the iterative performance of the formulations and solvers are shown on the left of Fig. 4, where the runtime for iteratively solving the MoM system has been taken from the fastest iterative solver for each formulation. For this particular comparison, unrestarted GMRES was not considered, in order to make a fair comparison among formulations. Indeed, for every formulation, GMRES is optimal; and, as a consequence, it outperforms all other Krylov methods in terms of



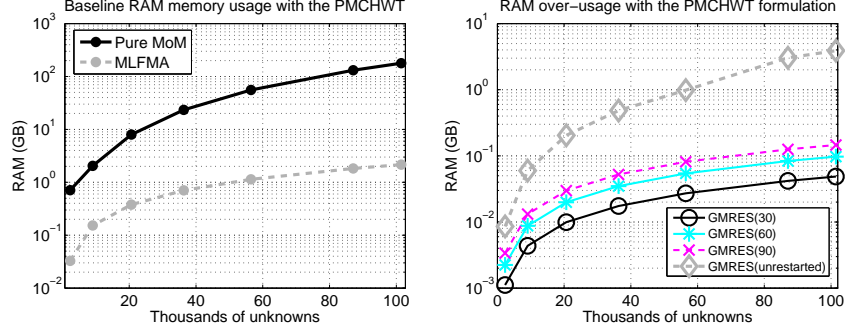


Figure 3: RAM memory usage with GMRES: (left) baseline memory usage in pure MoM and in MLFMA, without considering iterative solver requirements; (right) RAM consumption overuse for different GMRES versions with varying restarts.

iterative performance. However, as show above, unrestarted GMRES should be only used when memory requirements are not a concern.

Fig. 4 also includes on the right, for comparison purposes, the time required to fill in the full MoM matrix before the iterative solver begins. The time for filling the MoM matrix is slightly smaller with the PMCHWT and the CTF because, in these formulations, it is possible to avoid the integration of the Green's function gradient which appears in operator  $L_i$  in Eq. (3). This simplification on the integral requires employing the two-dimensional version of the divergence theorem, which cannot be invoked for the formulations that involve normal equations, due to the vector product by  $\hat{\mathbf{n}}(\mathbf{r})$ .

As shown in Fig. 4, the restarted GMRES(90) solver is the best choice for all the formulations, except for the tangential formulations PMCHWT and CTF, where TFQMR provides the best iterative behavior. Another important comment on Fig. 4 is that the MNMF formulation provides the best performance when solved iteratively.

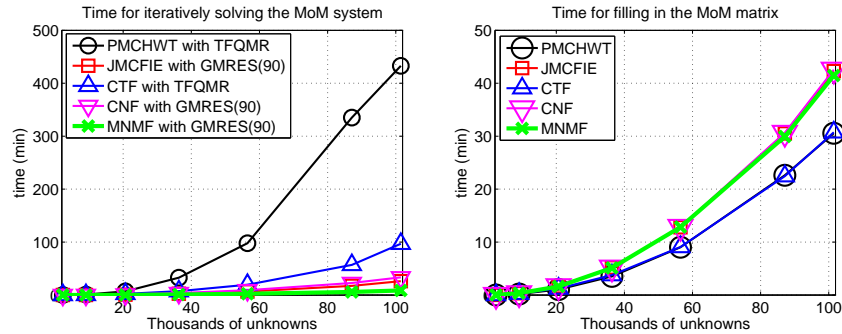


Figure 4: Runtimes for simulating the  $\lambda_0/2$  radius gold sphere: (left) runtime for iteratively solving the MoM system when the iterative method with the fastest convergence is chosen for each SIE formulation; (right) total runtime for filling in the MoM matrix.

Since, as discussed above in Subsection 4.1, the PMCHWT provides the best accuracy when dealing with plasmonic problems, we chose to represent in Fig. 5 the iterative performance of this formulation for each solver, including GMRES without restart. Even though important previous works such as [15] indicate that the PMCHWT is unfit to be solved by an iterative method, the scenario considered in this paper is different and not directly comparable to [15]. The recently introduced preconditioning scheme in [30], not yet available at the time of the report in [15], was

considered for the simulations in this work, as justified above.

As can be seen in the graphical results, CGS and TFQMR are the best choices for the PMCHWT, if unrestarted GMRES is excluded owing to memory requirements. Fig. 5 also shows that the number of matrix-vector multiplications (MVMs) required by each iterative solver is proportional to the runtime. It is worth stating that all the solvers require two MVMs in each iteration, except for GMRES (all versions), which requires only one MVM per internal iteration. The number of MVMs is thus a machine-independent parameter that can be easily used to compare the total processing time associated with each one of the formulations and solvers.

Let us note that for the formulations other than the PMCHWT all the solvers display a similar iterative behavior, as seen in Fig. 6.

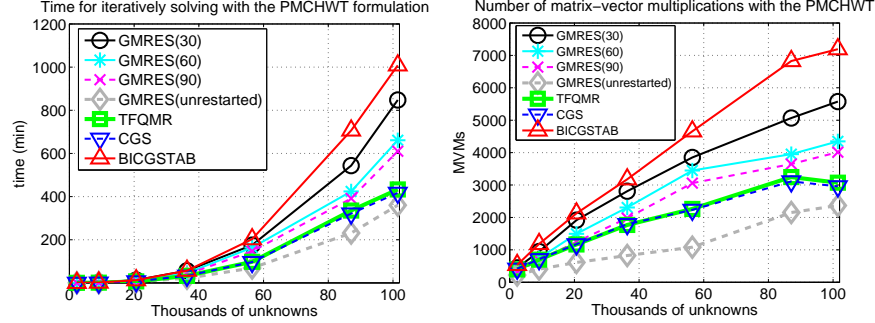


Figure 5: Iterative performance for the PMCHWT formulation in the considered Mie scattering problem: (left) time for solving the MoM system; (right) number of matrix-vector multiplications.

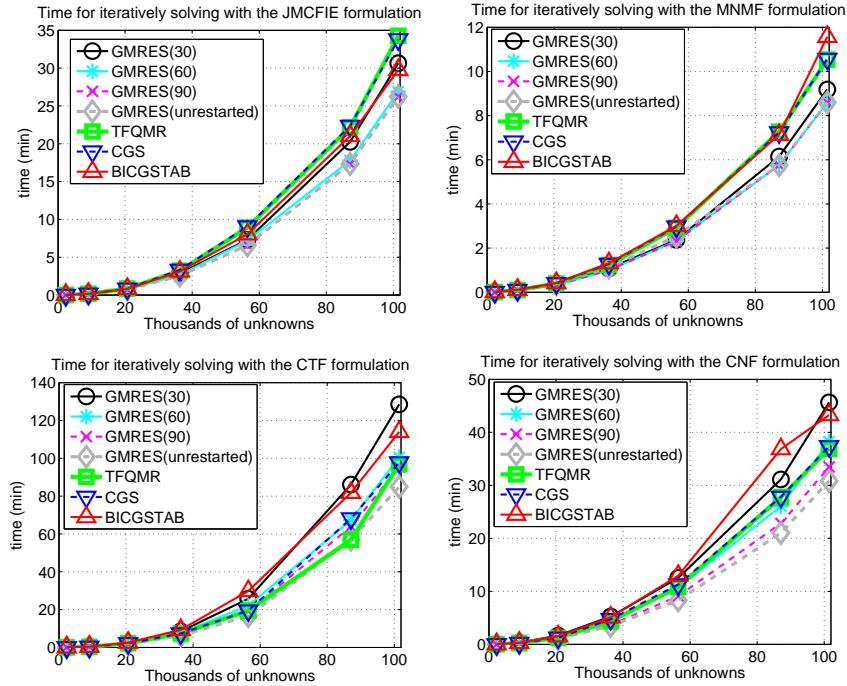


Figure 6: Iterative performance for the JMC FIE, MNMF, CTF and CNF formulations.

Fig. 7 shows the relative residual as a function of the number of MVMs (matrix-vector multiplications) for the PMCHWT formulation for 57,000 unknowns –other mesh sizes exhibit an analogous

behavior—. As expected, GMRES without restart exhibits the fastest residual decay. Regarding data from the CGS and BICGSTAB solvers in Fig. 7, it can be noted that the time-dependent residual variances are very high. A similar erratic residual evolution can be also observed in the iterative solvers included in commercial codes like FEKO; nonetheless, the important fact is that the time-averaged minimum residual is reduced. As a matter of fact, it must be clear from Fig. 7 that, for instance, both TFQMR (non-erratic) and CGS (erratic in this scenario) exhibit a similar time-averaged minimum residual evolution and they converge at a similar rate.

Returning to Fig. 7, it is clear that for the cases where memory-expensive unrestarted GMRES is not an option, TFQMR and CGS provide a better convergence than GMRES(90) in the PMCHWT, for the residue tolerance of  $10^{-6}$ ; however, this will not be the case if a higher residual error is defined to achieve convergence. This observation regarding the PMCHWT formulation is particularly important because, when programming a single SIE formulation in a MoM code, the PMCHWT is generally chosen as the preferred formulation due to the following reasons: 1) it is the easiest to implement, as it does not require using the normal vector directions; and 2) it is employed in commercially available software like FEKO [35] and in publications about simulating plasmonic bodies such as [8].

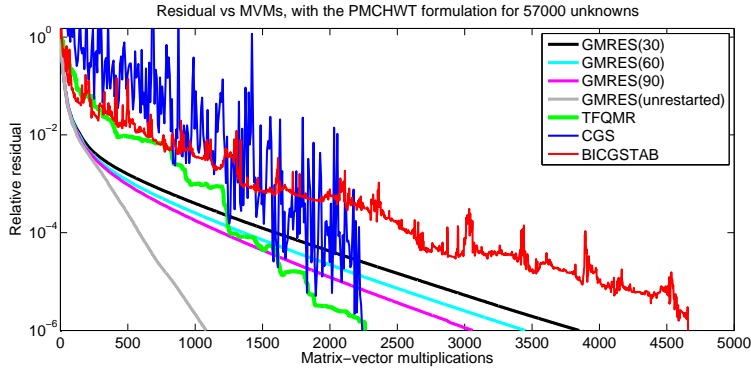


Figure 7: Relative residue versus matrix-vector products in the PMCHWT formulation, for each iterative solver.

In order to extract valid general conclusions about the iterative runtimes, we also analyzed spheres made of silver and aluminum and we tested more elaborated geometries at different optical frequencies, such as a nanoantenna made of aluminum. These general conclusions must be understood as valid under the simulation parameters justified at the beginning of Section 4. The use of alternative preconditioners would certainly be an important point to consider in order to obtain more general conclusions; however, these preconditioners would have required a noticeably extended explanation out of the scope of this paper. Likewise, despite the fact that higher-order basis functions are not indispensably required to perform accurate plasmonic simulations [20], we understand that they could be interesting for further research. Additionally, sharp edges and corners are not considered in the present work when extracting general conclusions. It is well known that sharp edges and corners are not typically found in real plasmonic structures chemically generated in a laboratory setup [11]; however we believe that additional research should be pursued in the future regarding plasmonic edges in both experimental and theoretical realms. Finally, following a similar approach used in the literature [18], we have only represented the most relevant cases when dealing with qualitative conclusions. The results corresponding to the spheres made of silver and aluminum, as well as simulations varying the nanoantenna plasmonic composition, are not shown for the sake of simplicity, as these results have proved to be qualitatively equivalent to the shown ones.

The considered real plasmonic Yagi-Uda nanoantenna for the emission of light at  $\lambda_0 = 570$  nm, whose emitted near field pattern obtained with our code is represented in Fig. 8, has been designed in [5]. See this reference for a detailed description of the antenna dimensions, which have been optimized for high optical directivity at the operating frequency. The optical antenna is made of aluminum ( $\epsilon_r = -38 - j10.9$  at the simulation frequency), and it enhances the emission of a single fluorescent chemical molecule modeled as a classical hertzian dipole along direction  $\hat{\mathbf{y}}$ , 4 nm above the feed element. In our particular simulation, the *equivalent* dipole has length 8 nm and its electric current is 1 nA.

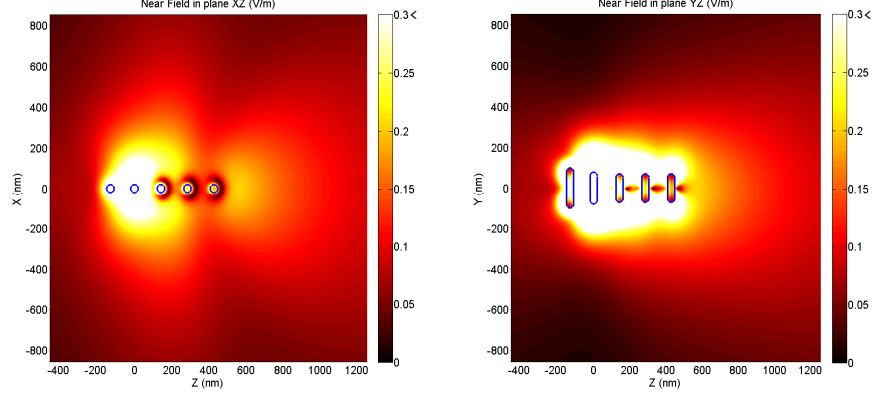


Figure 8: Near field pattern emitted by the aluminum nanoantenna designed in [5]. The represented near-field magnitudes were obtained with the PMCHWT formulation for a mesh size  $\ell_{discr} = \lambda_0/40$ .

As inferred from Fig. 9, iterative performance in the nanoantenna problem is similar to that previously presented for the Mie scattering problem when the sphere was simulated. The absolute runtimes versus the number of unknowns vary when compared to the Mie scattering, but the qualitative differences among the four iterative solvers remain very similar, which allows extracting some general conclusions for a wide variety of tests involving plasmonic media. These conclusions are summarized in the next section.

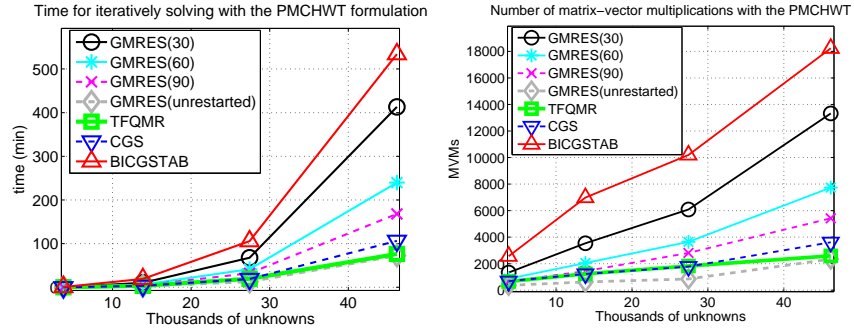


Figure 9: Iterative performance for the PMCHWT formulation when the aluminum nanoantenna in [5] is simulated: (left) time for solving the MoM system; (right) number of matrix-vector multiplications.

## 5 Conclusions

Four well-documented iterative solvers have been applied to five widespread MoM-SIE approaches in the analysis of nanostructures made of plasmonic materials. In order to increase the reproducibility of the results in this paper, we have provided in-depth details about all the parameters involved in our code implementation (numerical integration rule, mesh type, computational precision, etc.), and we have explained the reasons and convenience for the election of such parameters. These thorough explanations on code implementation are usually not described in detail, and they affect the reproducibility of results in previous papers about iterative solvers in computational electromagnetics. Our observations are summarized as follows:

1. In the case where memory requirements are not an impediment according to Eq. (5), then the best election for all the formulations is always GMRES without restart.
2. Whenever accuracy is required in the results, then the best choice is to use the PMCHWT formulation, because this formulation has the smallest error in all the analyzed plasmonic problems; however, the PMCHWT has a poor iterative performance.
3. If the PMCHWT is chosen and unrestarted GMRES cannot be applied due to memory requirements, then the best iterative solvers are TFQMR and CGS, because they may provide a much better performance, for the standard double-precision residue tolerance of  $10^{-6}$ , than the rest of the analyzed solvers. TFQMR and CGS are also the best choices for the CTF formulation (although in this case the effect is less noticeable).
4. The fastest convergence in plasmonics belongs to the MNMF and JMCIE formulations. For these formulations, importantly, however, the choice of the iterative solver has a small impact on the total runtime.
5. For a given fixed residual tolerance and iterative solver, the choice of the SIE formulation has a direct impact on the simulation error and also an impact on the runtime. Nonetheless, for a fixed residual tolerance and SIE formulation, the choice of the iterative solver has a negligible impact on the simulation error but, in general, a relevant impact on the total runtime.

In this paper, we did not only confirm conclusions in items 1 and 2 above –also approached in [16] and [20] respectively– but we have also extended our analysis to extract the remaining conclusions on the list above. These novel conclusions, together with the quantified results in the presented figures, are relevant contributions from this work to the know-how literature of computational electromagnetics in the context of plasmonic problems.

## Acknowledgement

The authors especially thank the company *Appentra Solutions*, developers of the automatic parallelizing source-to-source compiler *Parallware* ([www.appentra.com](http://www.appentra.com)), for assisting us in the analysis and parallelization of some parts of our C codes.

## Funding

This work is partially supported by the Spanish National Research and Development Program under project TEC2011-28683-C02-02, by the Spanish Government under project TACTICA, by the European Regional Development Fund (ERDF), and by the Galician Regional Government under

agreement for funding AtlantTIC (Atlantic Research Center for Information and Communication Technologies).

## References

- [1] Schuller JA, Barnard ES, Cai W, Jun YC, White JS, Brongersma ML. Plasmonics for extreme light concentration and manipulation. *Nature Mater.* 2010;9:193–204.
- [2] Maier SA. *Plasmonics: fundamentals and applications*. Springer; 2007.
- [3] Taminiau TH, Stefani FD, Segerink FB, van Hulst NF. Optical antennas direct single-molecule emission. *Nature Photon.* 2008;2:234–237.
- [4] Bharadwaj P, Deutsch B, Novotny L. Optical antennas. *Adv in Opt Photon.* 2009;1:438–483.
- [5] Taminiau TH, Stefani FD, van Hulst NF. Enhanced directional excitation and emission of single emitters by a nano-optical Yagi-Uda antenna. *Opt Express.* 2008;16:10858–10866.
- [6] Sugita T, Yanazawa K, Maeda S, Hofmann HF, Kadota Y. Radiation pattern of plasmonic nano-antennas in a homogeneous medium. *Opt Express.* 2014;22:13263–13268.
- [7] Filer R, Słowik K, Straubel J, Lederer F, Rockstuhl C. Nanoantennas for ultrabright single photon sources. *Opt Lett.* 2014;39:1246–1249.
- [8] Kern AM, Martin OJF. Surface integral formulation for 3D simulations of plasmonic and high permittivity nanostructures. *J Opt Soc Am A.* 2009;26:732–740.
- [9] Taboada JM, Rivero J, Obelleiro F, Araujo MG, Landesa L. Method-of-moments formulation for the analysis of plasmonic nano-optical antennas. *J Opt Soc Am A.* 2011;28:1341–1348.
- [10] Solís DM, Taboada JM, Obelleiro F, Landesa L. Optimization of an optical wireless nanolink using directive nanoantennas. *Opt Express.* 2013;21:2369–2377.
- [11] Solís DM, Taboada JM, Obelleiro F, Liz-Marzán LM, García de Abajo FJ. Toward ultimate nanoplasmonics modeling. *ACS Nano.* 2014;8:7559–7570.
- [12] Draine BT. The discrete-dipole approximation and its application to interstellar graphite grains. *Astrophys Journal.* 1988;333:848–872.
- [13] Hao F, Nehl CL, Hafner JH, Nordlander P. Plasmon resonances of a gold nanostar. *Nano Lett.* 2007;7:729–732.
- [14] Jin J. *The finite element method in electromagnetics*. Wiley; 2002.
- [15] Ylä-Oijala P, Taskinen M, Järvenpää S. Surface integral equation formulations for solving electromagnetic scattering problems with iterative methods. *Radio Sci.* 2005;40:RS6002.
- [16] Barsan V, Lungu RP. Trends in electromagnetism - from fundamentals to applications. In: *Tech*; 2012. Chapter 7, Fast preconditioned Krylov methods for boundary integral equations in electromagnetic scattering; p. 155–176.
- [17] Ergül Ö, Gürel L. Comparison of integral-equation formulations for the fast and accurate solution of scattering problems involving dielectric objects with the multilevel fast multipole algorithm. *IEEE Trans Antenn Propag.* 2009;57:176–187.

- [18] Araújo MG, Taboada JM, Rivero J, Obelleiro F. Comparison of surface integral equations for left-handed materials. *Prog Electromagn Res.* 2011;118:425–440.
- [19] Ergül Ö. Fast and accurate analysis of homogenized metamaterials with the surface integral equations and the multilevel fast multipole algorithm. *IEEE Antenn Wirel Propag Lett.* 2011;10:1286–1289.
- [20] Araújo MG, Taboada JM, Solís DM, Rivero J, Landesa L, Obelleiro F. Comparison of surface integral equation formulations for electromagnetic analysis of plasmonic nanoscatterers. *Opt Express.* 2012;20:9161–9171.
- [21] Saad Y, Schultz M. Gmres: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J Sci Statist Comput.* 1986;7:856–869.
- [22] Kelley CT. *Iterative methods for linear and nonlinear equations.* Society for Industrial and Applied Mathematics; 1995.
- [23] Ergül Ö. Accurate and efficient solutions of electromagnetics problems with the multilevel fast multipole algorithm [dissertation]. Ankara (Turkey): Bilkent University; 2009.
- [24] Bohren CF, Huffman DR. *Absorption and scattering of light by small particles.* John Wiley; 1983.
- [25] Gomez-Sousa H, Rubiños-Lopez O, Martinez-Lorenzo JA. Junction modeling for piecewise non-homogeneous geometries involving arbitrary materials. 2014 International Symposium on Antennas and Propagation; 2014 July 6-11, Memphis, Tennessee, USA. IEEE Antennas and Propagation Society (AP-S); 2014. p. 2196–2197.
- [26] Kim OS, Meincke P, Breinbjerg O. Method of moments solution of volume integral equations using higher-order hierarchical Legendre basis functions. *Radio Sci.* 2004;39:RS5003.
- [27] Taboada JM, Araújo MG, Bértolo JM, Landesa L, Obelleiro F, Rodríguez JL. MLFMA-FFT parallel algorithm for the solution of large-scale problems in electromagnetics. *Progr in Electromag Research.* 2010;105:15–30.
- [28] Teran JM. Givens QR and GMRES. Math 270C course notes. Mathematics Department, UCLA; 2011 [cited 2015 Sep 7]. Available from: [https://www.math.ucla.edu/~jteran/270c.1.11s/notes\\_wk2.pdf](https://www.math.ucla.edu/~jteran/270c.1.11s/notes_wk2.pdf)
- [29] Gibson WC. *The method of moments in electromagnetics.* 2nd ed. Chapman & Hall / CRC; 2014.
- [30] Landesa L, Araújo MG, Taboada JM, Bote L, Obelleiro F. Improving condition number and convergence of the surface integral-equation method of moments for penetrable bodies. *Opt Express.* 2012;20:17237–17249.
- [31] Geuzaine C, Remacle J-F. Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. *Internat J for Num Meth in Eng.* 2009;79:1309–1331.
- [32] Araújo MG, Taboada JM, Rivero J, Solís DM, Obelleiro F. Solution of large-scale plasmonic problems with the multilevel fast multipole algorithm. *Optics Lett.* 2012;37:416–418.
- [33] The Appentra team. Parallware: automatic parallelization of sequential codes [computer software]. A Coruña (Spain): Appentra Solutions SL; 2015. Available from: <http://www.appentra.com/products/parallware/>

- [34] The OpenMP team. OpenMP [application programming interface (API)]. OpenMP Architecture Review Board (<http://www.openmp.org>); 2015.
- [35] EM Software & Systems - S.A. (Pty) Ltd. Modelling of dielectric materials in FEKO. Technical report; 2005 [cited 2015 Sep 7]. Available from: [https://www.feko.info/about-us/quarterly/FEKO\\_Quarterly\\_Mar\\_2005.pdf](https://www.feko.info/about-us/quarterly/FEKO_Quarterly_Mar_2005.pdf)